Example Based Single-Image Super Resolution

A Project Report

submitted by

RAJESH KAKARLA

in partial fulfilment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2015

THESIS CERTIFICATE

This is to certify that the thesis titled **Example Based Single-Image Super Resolution**, submitted by **Rajesh Kakarla**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Devendra Jalihal
Research Guide
Professor
Dept. of Electical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 6th May 2015

ACKNOWLEDGEMENTS

I take this opportunity to express my deepest gratitude to my project guide Dr. Devendra Jalihal for his valuable guidance and motivation throughout the project. I am very grateful to him for providing his valuable time to guide me during the project. His guidance has helped me in completing my work, writing this thesis.

Besides my guide, I would like to thank Dr. R.Aravind for his thoughts and help on the Project which has helped me during my project work. I would also take this opportunity to thank Dr. A.N.Rajagopalan for organizing 'Image Signal Processing' course so well. His lab assignments after the theory classes gave us hands on experience with the techniques used in image processing. I would also like to thank the Department itself for providing such good facilities which were crucial in my work.

I would like to thank my friends Faizan, Bharath, Naresh, Rajkumar for their encouragement and help through out the project and for giving me good company throughout my stay in IIT.

Finally I would like to thank my family. I am indebted to them forever for their never ending support and for giving me the freedom to make the decisions in my life.

ABSTRACT

KEYWORDS: Markov Random Field

The word "Super-Resolution" refers to the class of techniques of obtaining higher-resolution image from Lower resolution one. Super resolution techniques handles the issues of aliasing, blurring, noising. Most of the researchers on super resolution use Multi-Frame methods where non redundant information of the pixels is found out by the sub-pixel shifted low resolution observations. Recently super resolution from single image has also started.

Many different approaches are being used to for solving this problem, one of such attempts is probabilistic approach. This is one of the very early probabilistic methods that was suggested. This approach was suggested by Freeman *et al.* (2002), where we seek to incorporate the relationship between the neighbours of the image pixels which brings smoothness to the image and increases image quality to the eye

Here we develop a training set dictionary for both low and high resolution images and use them for creating plausible high frequency details in enlarged images. The algorithm develops a Markov Random Field framework to probabilistically relate the high resolution and low resolution image. It uses an iterative Belief Propagation algorithm, which usually converges quickly.

The second approach is a one-pass algorithm that uses the same local relationship information as the Markov network. It's a fast, approximate solution to the Markov network model. These methods preserve fine details, such as edges, generate believable textures, and can give good results even after zooming multiple octaves.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

LR	Low Resolution
HR	High Resolution
SR	Super Resolution
MMSE	Minimum Mean Square Error
MMSE	Maximum A Posteriori
HMM	Hidden Markov Model
BP	Belief Propagation
MRF	Markov Random Field

NOTATION

Ψ	Compatibility Matrix
Φ	Compatibility Function
m_{ij}	Message from node i node j

CHAPTER 1

INTRODUCTION

Tsai and Hunag were the first to consider the problem of obtaining a high-quality image from several down-sampled and translationally displaced images in early 1980's. Their data set consisted of terrestrial photographs taken by satellites. They modelled the photographs as aliased, translationally displaced versions of a constant scene. Their approach consisted in formulating a set of equations in the frequency domain, by using the shift property of the Fourier transform.

1.1 Motivation

The high-resolution images not only give people good visual delightment but also provide with important details for pattern recognition and image analysis. However, during the process of the image capture sometimes, the collected images are often with low-resolution due to the poor equipment performance and environment.

Now a days, people demand the images with higher and higher resolution in their work and life. If the resolution of the images is raised in the most direct method which applies the high-resolution sensor, people have to pay high prices for the precision optics that is unable to apply on many occasions or in large-scale deployment. Obviously, this method can't be applied to the ordinary commercial business. Therefore, it's hard to raise the resolution in terms of hardware.

Shot Noise

Another problem arises if the resolution increases continuously, increasing the spatial resolution reduces the pixel size, i.e increase the number of pixels per unit area, which decreases the amount of light getting into each pixel. Due to random nature of light, the shot noise in each pixel increases which makes the image noisy.

Under this context, more and more researchers have begun to apply the signal processing as the method to break through the bottleneck of the hardware in order to acquire high-resolution images. This method is called image super-resolution technology.

To be specific, in the areas of public safety and counter-terrorism, due to the limitation from the cost and technology or the environment, the resolution of videos or images are not high, so it's hard to discover the latent danger from these videos or images. However, people can capture some abnormal behaviours from high-resolution images that are generated by super-resolution image reconstruction. The high-resolution contains a great deal of important detailed information that is useful. For example, the high-resolution medical image is helpful for the doctor to make a right diagnosis.

1.2 Resolution

The concept of the resolution in image processing is related to the amount of information embedded in an image. One of the technical we need to be clear about is the term 'Pixel' which was invented from "Picture Element" word.

One other technical term which is of interest to us is the term 'Resolution'. Let us see different types of it and understand it in a better detail, so that we will be clear what we are trying to do when we say Super Resolving Images. The word 'Resolution' which is a metric that evaluates the quality of the image. In optics the resolution is determined the response of the system to various different spatial frequencies. The term is classified into different types, some of which are:

- *Spatial Resolution:* This resolution refers to the spacing of the pixels in an image. The higher the spatial resolution the higher the perception of sharp details and colour transitions in an image. If suppose an image having higher level of details is not represented by dense set of pixels, then the resultant image suffers from aliasing arti-facts.
- *Brightness Resolution:* This resolution refers to the gray scale colour range of each individual pixel. Generally the brightness of colour value of each individual pixel is defined by a group of bits. In case of coloured image we use 24 bits 8 for each of one red, blue, green.
- *Spectral Resolution:* This is the spectral or frequency resolving power of a sensor and it is defined as the smallest solvable wavelength difference by the sensor.

- *Temporal Resolution:* This refers to the rate of frames or the number of frames that are captured per second. The higher the frame rate the less the smearing affects due to the movements in the scene.

Here in this thesis when we say resolution we always refer to the **Spatial Resolution** or pixel resolution.

1.3 Structure of the Thesis

The Chapter on **Introduction** gives the research background and importance, motivation for this project, arrangement of the tasks and the research scoping are introduced.

The Chapter on **Background** is about the background of the image super-resolution. In this chapter, we introduce the past and current researches on the image super-resolution technology. In addition, the two mainstream classification of super-resolution methods are reviewed.

The Chapter on **Single Image Super Resolution** introduces the study on the example-based super-resolution method. In this chapter, we describe the probabilistic approach used for implementation and different algorithms that are used as a solution for this problem.

The Chapter on **Implementation Details**describes the pseudo code of the generating training dictionaries and Super Resolution Algorithm and describes the importance of training data set and parameters used in implementing the algorithm.

The Chapter on **Experimental Results** shows the output of our algorithm and compares it with interpolation and independent patches approach. Influence of various parameters on the output is also seen in this Chapter

The Chapter on **C++ Implementation** discusses about C++ code in Open CV, all the modules and functions used in the code are explained, problems that came up while implementing in Open CV are also mentioned.

The Chapter on **Future work and Conclusions** discusses about some areas that can be studied and improved which can improve the quality of the output image, and also discusses about the standard methods that are used for image quality evaluation.

CHAPTER 2

Background

In order to comprehensively analyze the super-resolution image reconstruction problem, we need to build a model that relates the original high-resolution image to the observed low-resolution to simulate the process of obtaining the low-resolution image from the high-resolution image. This model is called observation model. It's critical to build a precise system model for the image super-resolution reconstruction technology. The common observation model for images is shown in Fig 2.1

2.1 Observation Model

The model shown in Fig 2.1 can be expresses with the following notations

$$y_k = D\,B_k\,M_k\,x + n_k,\;\; 1 \leq k \leq P$$

where P denotes the number of low resolution images x, y_k and n_k denote the high resolution image, the kth observed low-resolution image and the noise vector respectively. The matrices of D, B_k and M_k denote the sub-sampling matrix, blur matrix and the warp matrix. So this equation can be simplified by

$$y_k = Hx + n_k$$

Super Resolution technology in the early research only stood for reconstruction method based on multiple images, while the enhancement based on single image was called interpolation. However, presently researchers have developed methods for even single image super resolution which perform better than interpolation and we will see one of them in later chapters of the Thesis.

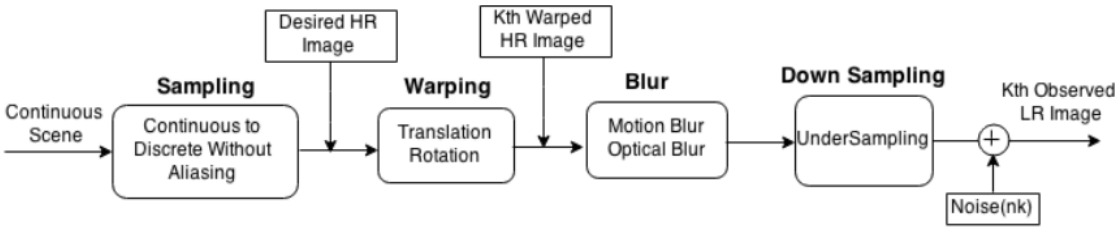


Figure 2.1: Observation Model Relating LR and HR images

2.2 Different Approaches for Super Resolution

Many approaches under super-resolution processes use more than one image and is computationally complex. But these methods reconstruct the high resolution image quite accurately. Several approaches are being used to solve this problem and the approach depends on the kind of application and the data we have. Broadly these methods can be classified into two types apart from the standard interpolation techniques.
Fig 2.2 shows classification of methods that are used in super resolution.

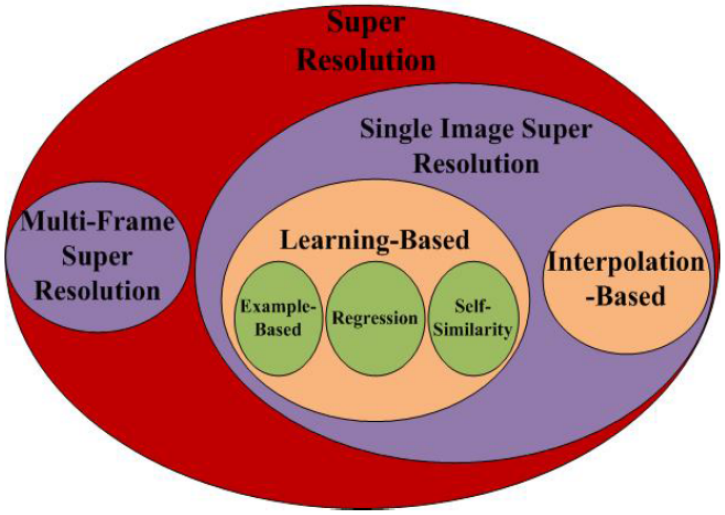


Figure 2.2: Classification of Super Resolution Methods

2.2.1 Aggregating from multiple frames

This is generally termed as multi-frame super resolution. In this context of super resolution for images, it is assumed that we can combine several LR images into a single High

Resolution image, we are decreasing the time resolution, and increasing the spatial frequency content. But there must be some variation between them such as translational motion parallel to the image plane, some other type of motion (rotation, moving away or toward the camera), or different viewing angles.

Let us see a simple example of how we can construct the original signal from a set of sampled,translated versions of signal. For simplicity let us consider a 1D signal and try to construct it from given sampled and translated versions of the signal.

Let $x(t) = \sin(2\pi f_0) + \sin(2\pi f_1)$, where $f_0 = 4Hz$, $f_1 = 4.3Hz$. Now $x[n] = x(nT_s)$ containing N samples, such that $\frac{1}{NT_s} < f_1 - f_0$ and $T_s = 0.1s$.

Now let us generate two sequences which are translated and sampled versions of the original signals,

$$y_1 = DW_1x$$

$$y_2 = DW_2x$$

where W is the warping matrix and D is the sampling matrix, here let us take W_1 to be identity, W_2 to be translating matrix with $\delta x = 0.3$ and D is the down-sampling matrix by a factor of 2.

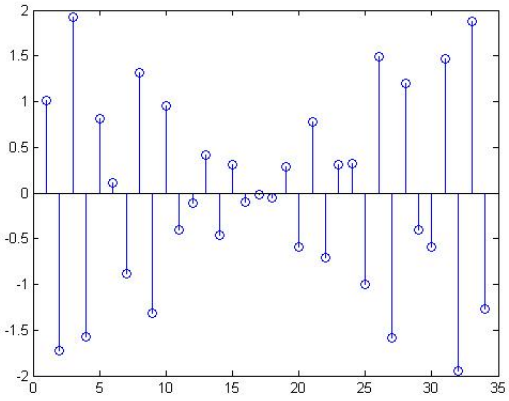
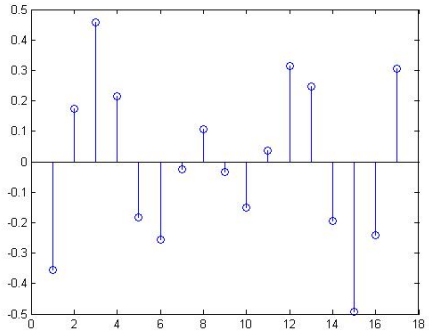
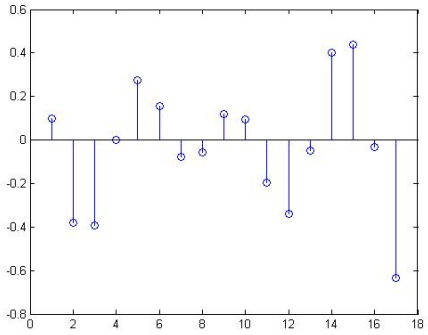


Figure 2.3: Original Signal $x[n]$



(a) Downsampling by 2 and No Translation



(b) Downsampling by 2 and Translation by 0.3

Figure 2.4: Sampled and Translated versions of Original Signals

Matrix D is down-sampling matrix with one in corresponding positions and matrix W is filled with elements δ and $1 - \delta$ correspondingly and original signal is obtained by inverting these matrices.

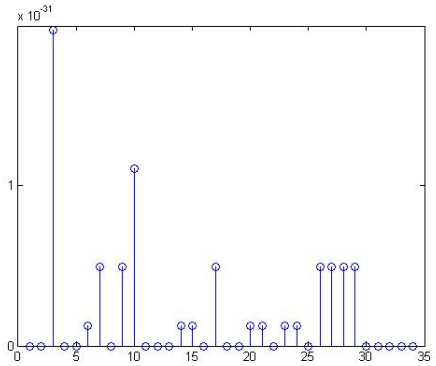


Figure 2.5: Error obtained from Estimated signal

2.2.2 Single-frame super-resolution

In this approach, there are different methods that can be used for super resolution. One common method that is very straight forward is interpolation techniques.Image interpolation techniques can also increase the resolution of an image and give the same amount of pixels that super-resolution can give. But there is quite a lot of difference among them.

Image interpolation finds the unknown pixel values by fitting the known pixels to a polynomial. Bi-linear and bi-cubic interpolations are examples to this and they are

widely used since they are computationally cheap. But the downside to this is that image interpolated are always smooth i.e, they do not have the high frequency details,they introduce arti-facts and blur edges. Super resolution on the other hand makes use of additional information such as internal and external statistics of image, image prior etc.

In Fig. 2.6a we can observe the blur edges where as in Fig. 2.6b the high frequency details are much better. Some of the other methods that are used as tools for super resolution are Recursive Least Squares, Probabilistic Methods, Projection on to Convex Sets. So we try using learning based techniques for estimating the higher frequency details that are not present in the interpolated images. Patch redundancy, Training data base are some of the techniques used in learning based methods.

(a) Interpolated Output (b) Super Resolution Output

Figure 2.6: Difference between Interpolation and Super Resolution

Why should Learning Based approach work

In this Thesis we focus on Single frame resolution, as the prolificacy of the real world images is hard to gain control the algorithm should carefully learn the fine details from the Training data set. This approach uses Probabilistic methods and belief propagation algorithm is used as a solution. To see why this approach should work at all.

Consider that a collection of image pixels are special signals that have much lesser variability than a corresponding set of random variables. It is studied that these regularities account for early processing stages of visual systems. We exploit these regularities and use small pieces of training images for creating plausible high frequency details. Pre-processing of these small images are done for generalization without restricting them to specific class of training images so that we can generate correct high resolution information.

CHAPTER 3

Single Image Super Resolution Algorithm

3.1 Introduction

Freeman *et al.* (2002) proposed this method of example-based super-resolution algorithm. Firstly, the high-resolution images are applied to form the training set, which contains the high-frequency of the images and a group of sub-sample low-resolution images. Later on, in the training set, the knowledge about the high-frequency and the corresponding low-frequency is learnt, this knowledge is used to predict the details of other low-resolution images. The overview of their method as shown in the Fig 3.1.

The main goal of the Freeman *et al.* (2002) algorithm is to estimate the high-resolution components which are missing in the original image. The entire algorithm can be categorized in two main sections. First section is related to the training stage where the example patches are obtained from an external comprehensive database. The database learns correlations between low and high resolution image patches which form pairs.

Example-based method is working in such a way that two sets of data, which are employed as training patches to construct the new HR image, are produced. Recall the observation model from the previous section,using the same idea Training set is generated. In the second section, the super resolving algorithm is completed by utilizing a graphical framework (MRF or One-Pass algorithm) which helps to preserve the neighbour compatibility.

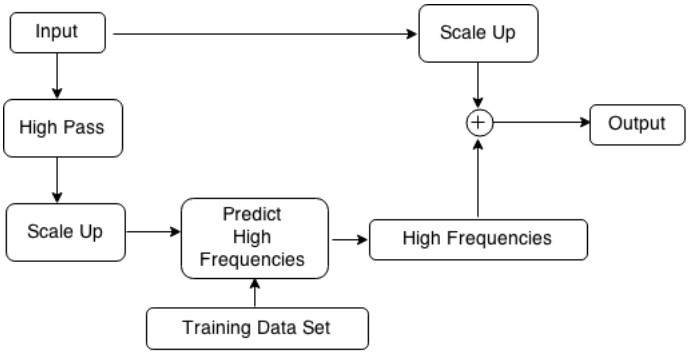


Figure 3.1: Overview of Freeman et al method

3.2 Generation of Training data set

For generation of the training set, we start from collection of high-resolution images as in Fig 3.3b and degrade each of them in a way matching to the degradation we plan to unmake in the images we later process. On the whole, we blur and sub-sample them to create a low-resolution image as in Fig 3.2 of one-half the number of original pixels in each dimension. We use the single octave algorithm recursively, to change resolution by higher factors.

For generating an image of the required number of pixels, we apply an initial numerical interpolation, such as cubic spline, to the low-resolution image, but it lacks high-resolution details as in Fig 3.3a. In our training set, we only need to store the differences between the image's cubic-spline interpolation and the original high-resolution image. Fig 3.2 and Fig 3.3b show low and high-resolution versions of an image. Over view of the algorithm is shown in Fig 3.4

Figure 3.2: Low Resolution Image

We store the HR patch corresponding to every possible low-resolution image patch. Though we restrict ourselves to apparently reasonable image information, we need to store a huge amount of information. So pre-processing of the images needs to be done to remove variability and make the training sets as generally applicable as possible.

(a) Interpolated version of LR Image

(b) High Resolution Image

The training set generated is used for training the model, here we believe that the following assumptions hold

- The highest spatial-frequency components of the low-resolution image shown in Fig. 3.3a are most important in predicting the extra details in Fig 3.3b. We filter out the lowest frequency components in Fig 3.3a so that we need not have to store example patches for all possible lowest frequency component values.
- The relationship between high and low-resolution image patches is essentially independent of local image contrast. We needn't store examples of that underlying relationship for all possible values of the local image contrast.

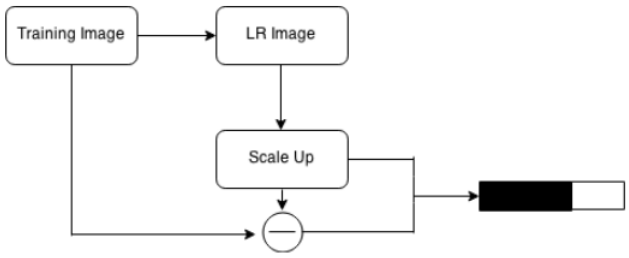
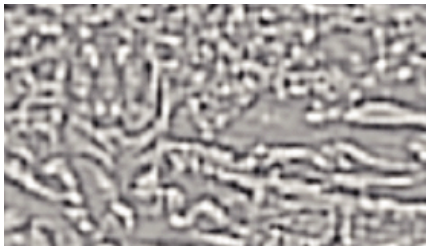
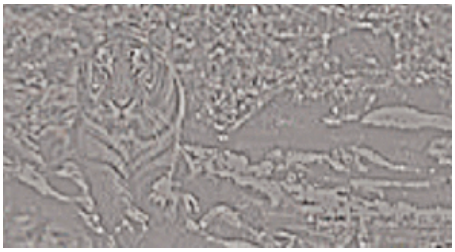


Figure 3.4: Overview of Training Data Set

We apply local contrast normalization, which we describe later on in the "Prediction" section. In Fig 3.5a and Fig 3.5b, we used the resulting bandpass filtered and contrast normalized image pairs for training. We undo the contrast normalization step when we reconstruct the high-resolution image.



(a) Bandpass filtered Interpolated Image



(b) High-pass filtered HR Image

3.3 Algorithm

We have already seen why this training approach should work for Super Resolution. We need to also know that the local image information alone are not sufficient to predict the missing high frequency details, so we just can't use the training patches themselves for Super Resolution. For a given input test image we apply the pre-processing steps, and break the input image into patches and look for the missing high frequency details. But such approaches don't work. The local patch alone isn't sufficient to estimate the plausible high frequency details.

Figure 3.6: Input patch of a test image

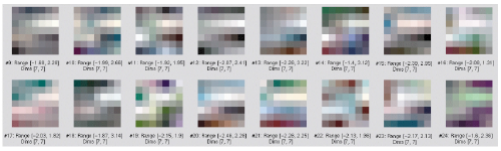


Figure 3.7: Closest 16 neighbours of the given input patch

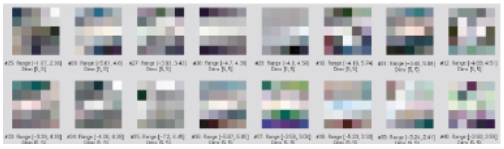


Figure 3.8: Corresponding selected High resolution patches

For a patch in the test image like in Fig 3.6, finding 16 closest training patches in the data base look like in Fig 3.7. Each of the individual patch looks similar to the input patch. But when we check their corresponding high resolution patches, they look fairly different from each other and look like Fig 3.8. So considering only the local features of the image will not suffice, we need to take into account the spatial neighbouring effects. Fig 3.9 shows us the estimated high frequency details of the given image.

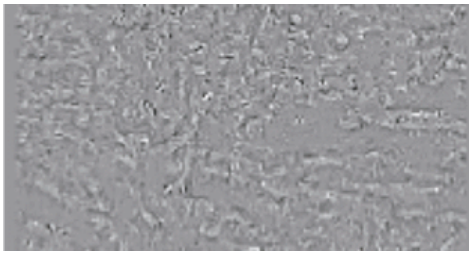


Figure 3.9: Estimated High frequencies

3.3.1 Markov Network

The spatial relationship between the patches is modeled using Markov Network. The circles in Fig 3.9 represent the nodes and the edges connecting them indicate the statistical dependencies between the nodes. The low resolution image patches be y, different states of the hidden nodes, x are the nearest neighbours.

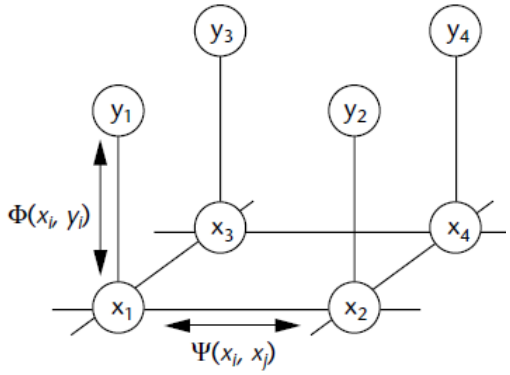


Figure 3.10: Markov Network indicating statistical dependencies between LR and HR patches

For this given network, the probability of any given high resolution choice for a given input patch is the product of all sets compatibility matrices 'Ψ' connecting to possible states of each pair of neighbouring hidden nodes and vectors 'Φ' relating each observation to hidden states

$$P(x/y) = \frac{1}{Z} \prod_{ij} \Psi_{ij}(x_i, x_j) \ \Phi_i(x_i, x_j)$$

Z is just a normalizing constant, the first product $\Psi_{ij}(x_i, x_j)$ is over all the neighbouring pairs of nodes i and j. y_i is the observed low resolution patch and x_i is the

estimated high resolution patch at node i.

For computing Ψ, we sample the input image nodes in a way that there is an overlap with each other by one or more pixels. In this region of overlap, the pixel values of agreeable neighbouring patches should correspond. Let $d(i,j)$ be the sum of squared differences between the patch candidates x_i and x_j in the region of overlap at nodes i and j. The compatibility matrix between nodes i and j is

$$\Psi_{ij}(x_i, x_j) = \quad e^{\frac{d_{ij}(x_i,x_j)}{2\sigma^2}}$$

Here sigma is the noise parameter. A similar penalization is used on differences between the observed LR patch y_i and the LR patch x_i found from the training data set for $\Phi(x_i, y_i)$.

The optimal high-resolution patches at each node is the set that maximizes the Markov network posterior probabilities. Here finding the accurate solution is computationally impracticable. So we use Belief Propagation Algorithm as an approximate solution for the Markov Network. The algorithm updates message m_{ij} from node i to node j, which have dimensionality of the state we want to estimate at node j. We use $m_{ij}(x_j)$ to show the component of vector m_{ij} corresponding to the patch x_j, the updating message from node i to node j is

$$m_{ij}(x_j) = \sum_{x_i} \ \Phi_{ij}(x_i, x_j) \prod_{k \neq j} \ m_{ki}(x_i) \ \ \Phi_i(x_i, x_j)$$

The sum is over all the patches x_i at node i and product is over all neighbours of the node i except for node j

3.3.2 Belief Propagation Algorithm

In our approach, training data provides candidate high-resolution explanations for the low-resolution image data. Modelling the image as a Markov network, Bayesian belief propagation quickly finds estimates for the most probable corresponding high-resolution explanation.

For given image data, y we seek to estimate the underlying scene, x. We first calcu-

late the posterior probability $P(x/y) = CP(x,y)$.Under two common loss functions, the best scene estimate, $\hat{x}$, is the mean with minimum mean squared error(MMSE) sense or the mode with the maximum a posterior(MAP) sense of the posterior probability.

For networks without loops, the Markov assumption leads to simple "message-passing" rules for computing the MAP and MMSE estimates.To derive these rules, we first write the MAP and MMSE estimates for x_j at node j by marginalizing (MMSE) or taking the maximum (MAP) over the other variables in the posterior probability.

$$\hat{x}_{jMMSE} = \int_{x_j} x_j dx_j \int_{all \;\; x_i \;\; i \neq j} P(x,y) \;\; dx \tag{3.1}$$

$$\hat{x}_{jMAP} \;\; = \;\; \underset{[x_j]}{argmax} \;\; \underset{[all \;\; x_i \;\; i \neq j]}{max} \;\; P(x,y) \tag{3.2}$$

For a Markov random field, the joint probability over the scenes x and images y can be written as

$$P(x,y) = \prod_{neighbouring \;\; i,j} \Psi(x_i, x_j) \;\; \prod_k \;\; \Phi(x_k, y_k) \tag{3.3}$$

where we have introduced pairwise compatibility functions Ψ and Φ, described below. The factorized structure of $Eq.3.3$ allows the marginalization and maximization operators of $Eq(3.1)$ and $Eq(3.2)$ to pass through compatibility function factors with unrelated arguments.

Consider the example network of three nodes, $x1$, $x2$, and $x3$, connected in a chain, with a corresponding y_i node attached to each x_i node. We have

$$x_{1MAP} = argmax_{x1} \, max_{x2} \, max_{x3} \quad P(x_1, x_2, x_3, y_1, y_2, y_3) \tag{3.4}$$

$$= argmax_{x1} \, max_{x2} \, max_{x3} \quad \Phi(x_1, y_1) \, \Phi(x_2, y_2) \, \Phi(x_3, y_3) \Psi(x_1, x_2) \, \Psi(x_2, x_3) \tag{3.5}$$

$$= argmax_{x1} \, \Phi(x_1, y_1) \quad max_{x1} \, \Psi(x_1, x_2) \, \Phi(x_2, y_2) \quad max_{x1} \, \Psi(x_2, x_3) \, \Phi(x_3, y_3) \tag{3.6}$$

Each line of $Eq.(3.6)$ is a local computation involving only one node and its neighbours. The analogous expressions for x_{2MAP} and x_{3MAP} use the same local calculations. Iterating those calculations lets each node j compute x_{jMAP} from the messages passed between nodes.

This works for any network without loops, $Eq.(3.2)$ and $Eq.(3.1)$ can be computed by iterating the following steps. The MAP estimate at node j is

$$\hat{x}_{jMAP} \;=\; \underset{[x_j]}{argmax}\;\; \Phi(x_j, y_j) \prod_k L_{kj} \tag{3.7}$$

where k runs over all scene node neighbours of node j. We calculate L_{kj} from

$$L_{kj} = \underset{[x_k]}{max}\;\; \Psi(x_k, x_j)\, \Phi(x_k, y_k) \prod_{l \neq j} \tilde{L_{lk}}\, dx_k \tag{3.8}$$

where $\tilde{L_{lk}}$ is L_{lk} from previous iteration. The initial $\tilde{L_{lk}}'s$ are 1. After at most one iteration of $Eq(3.8)$ per scene node variable $Eq(3.7)$ gives the desired optimal estimate x_{jMAP}. The MMSE estimate, $Eq(3.2)$ has analogous formulae with the max_{xk} of $Eq(3.8)$ replaced by $\int_{x_k}$, and argmax x_j of $Eq(3.7)$ replaced by $\int_{x_j} x_j$. For linear topologies, these propagation rules are equivalent to standard Bayesian inference methods, such as the Kalman filter and the forward-backward algorithm for Hidden Markov Models.

For a network with loops, the factorization of $Eq.(3.1)$ and $Eq(3.2)$ into local calculations doesn't hold. Finding the posterior probability distribution for a Markov network with loops is computationally expensive and researchers have proposed a variety of approximations. Recent theoretical work provide support for a very simple approximation: applying the propagation rules derived above even in the network with loops.

The condition of local optimality for the converged solution of the MAP algorithm is a strong one. For every subset of nodes of the network which form a tree, if the remaining network nodes are constrained to their converged values, the values of the sub-tree's nodes found by the MAP algorithm are the global maximum over that tree's nodes. These experimental and theoretical results motivate and justify applying the belief propagation rules even in a Markov network with loops.

3.3.3 One Pass Algorithm

In this method, the same markov network approach is used but this avoids the various steps in setting up and solving the Markov Random Field. Here the high resolution patch compatibilities are computed only for the for neighbouring patches that already got selected.

If the training data is pre-structured properly we will be able to match the local LR image data and select the compatible HR patch in one single task, i.e finding the nearest neighbour to a given input in the training set. At each node find the candidate set, find the compatibility functions between all the pairs of nodes and use belief propagation algorithm. Fig 3.11 shows us an overview of one pass algorithm.

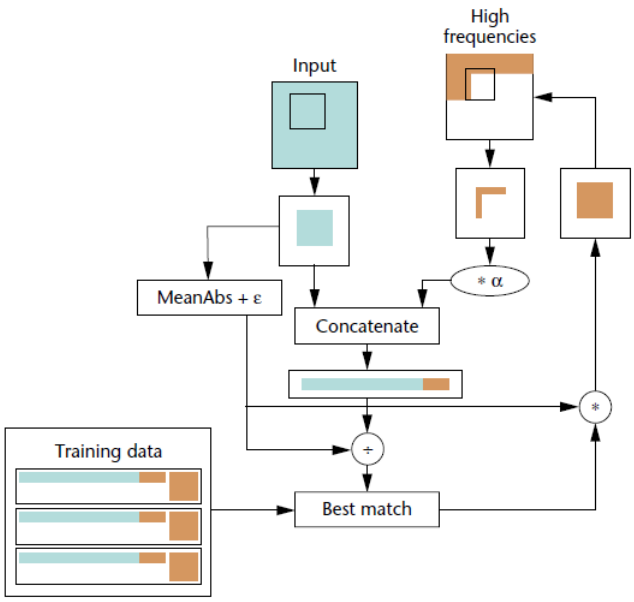


Figure 3.11: Overview of One Pass Algorithm

3.4 Prediction

For the given high frequencies in the input image, the algorithm needs to predict the missing frequencies that are missing from a zoomed-up image with interpolation. The output of the algorithm is the sum of high frequency estimates and the input.

The prediction of the high frequency details are based on two major requirements. Firstly the fine high frequency patch that needs to be replaced should come from training set image that has a similar resemblance of the low frequency patch. Along with that the high frequency patch selected should match with the edges of the patch with the coinciding pixels of the neighbouring patches. The first requisite is fulfilled by extracting a low frequency patch from the image we are trying to match in the training set, which has a pair of Low resolution and high resolution patches.

For the second requirement, we overlap the predicted patches at their boundaries. So when we search through the training data set, we use the high frequency samples previously predicted to select the best pair. We control the relative importance of matching the low frequency patch and matching the neighbouring selected high frequency patches by adjusting the factor α.

This algorithm functions with the assumptions that predictive relationship between the low and high resolution patches is independent of the local image contrast. So what we do is normalize the patch pairs by the average absolute value of the LR patch across the colour channels. The pixels that are present in the low frequency patch and the high frequency overlap are chained to form a lookup vector. Even the training set is stored in the same way, when we find a match, we undo the contrast normalization step on the high frequency patch selected and add it to the interpolated image to get the High Resolution output.

CHAPTER 4

Implementation Details

In this chapter we will see some of the modules that are implemented which needs certain attention and discuss about the importance of training set and parameters that are used in the implementation

4.1 Training Data Set

Algorithm 1 Generating Training Data Set

---

for every HR image (Fig 4.1a) in database **do**

down-sample it by required factor

interpolate to get corresponding LR image with same no of pixels (Fig 4.1b)

subtract this from the HR image & store the high frequency details (Fig 4.2a)

downsample the LR image further & interpolate it back to original size

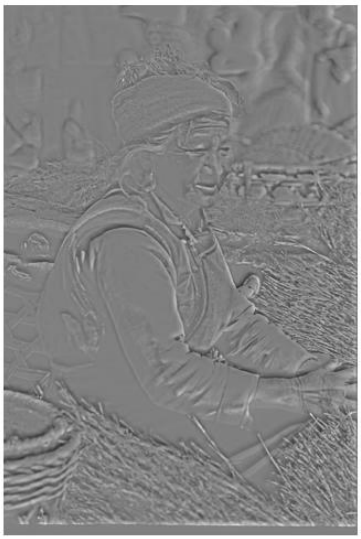
subtract it from LR image and store the low frequency details (Fig 4.2b)

create a LR HR patch dictionary from the stored image details

end for

(a) High Resolution Image

(b) Interpolated Image



(a) HR frequency details

(b) LR frequency details

4.2 Super Resolution Algorithm

In this method, the input LR image is up-sampled using bi-cubic interpolation. Thus, the up-sampled image, assume I_h^l , is enlarged with the desired scaling factor. Certainly, this image is blurred compared to the original version of itself. Afterwards, I_h^l is passed through a band-pass filter which filters the LF components while preserving the HF components in the image.

One of the major issues in this method is that the LR image, i.e. I_h^l, is divided into several small patches. This means that, the observation model which is made between LR and HR patches is local. Hence, it is possible that the chosen HR patch would not be compatible with its spatial neighbours. Consequently, the algorithm should not only search through the LR database to find the closest patch to the input image patch. Indeed, in order to have an accurate prediction the surrounding patches in HR database are needed to be taken into consideration. The results of the these closest patches is seen in Fig 3.7 in the previous chapter.

In Freeman *et al.* (2002) approach instead of applying MRC algorithm, One-Pass has been utilized. Although the one-pass algorithm is similar to Markov network, it

is faster in implementation because of considering only HR patch compatibilities. The example-based algorithm is expressed as following:

DATA: Input LR image I_l, training set (patches), interpolated image I_h^l

---

Algorithm 2 Freeman Algorithm

for each patch of I_l **do**

Search in LR patches and find NN no. of closest patches

Finding their corresponding HR patches

Applying Markov network for neighbouring information & find the best LR patch

Select its corresponding HR patch, add it with the patch in the interpolated image

Merge all the patches to form the resultant output Image

end for

As it was previously mentioned, the Freeman *et al.* (2002) method is one of the first approaches toward the example-based super resolution algorithms. Later works with the concept of example-based methods have made their efforts on eliminating some constraints introduced by Freeman et al algorithm. The most significant issue in example-based method is the requirement of a huge training data (high dimensional data) set to cover all possible patterns.

4.3 Search Algorithm

Our criteria for searching is the best match patch is $L2$ norm. For a low-resolution patch, it's necessary to inquire a training set containing hundreds of thousands patches. Due to the very high dimension of the search space finding the best match is computationally very costly. If the best matching patch is acquired by a brute force method, the computation cost is large.Here there is a trade-off between time for computations and best match. Here, we use the k-d tree to seek the nearest neighbour.

The k-d tree is a binary space partitioning tree structure produced for organizing points in a k-dimension space. In this structure, every non-leaf node can be thought of as implicitly half-spaces which the space is divided into two parts by a splitting hyperplane.

And the each half-space can be divided recursively in the same way, all the half-spaces are divided into the left and right sub-tree. The division of the k-d tree is conducted along the coordinate axis, and all the hyperplanes are perpendicular to the corresponding coordinate axis.

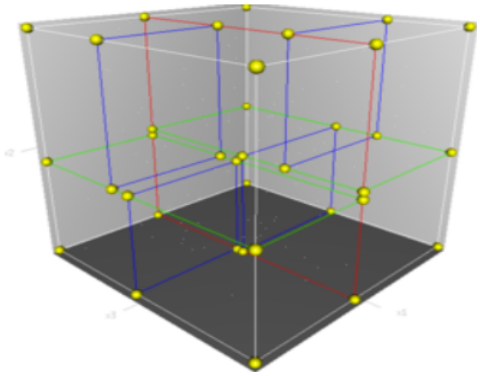


Figure 4.3: 3 dimensional KD Tree

The k-d tree is a fairly effective organization structure for the k-dimension data, a $3D$ model of which is shown in Fig 4.3. We use tree search to find the best match, this allows for speed computation with trading-off with the quality. As we don't get the true best match of the input patch, we use greedy downhill approach and try to get the match in the graph connecting approximate nearest neighbours in the training data set. In the one pass algorithm discussed before we connect each patch pair to its NN(32) approximate nearest neighbours.

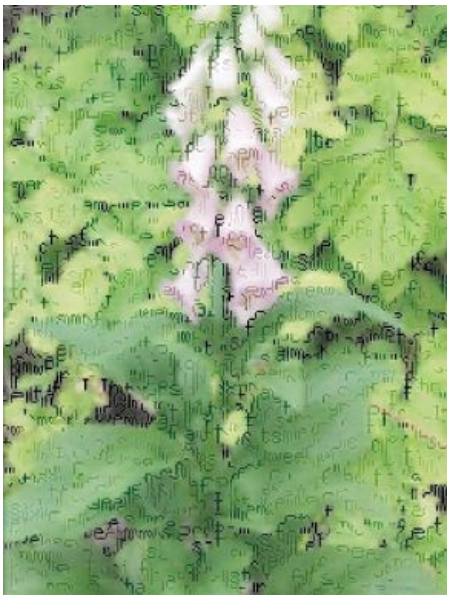
4.4 Training Set and Parameters

Although the training data set need not be very similar to the image we are trying to enlarge, it should be of similar image class, we enlarged the image in Fig 4.4a using a training set of images of text. The algorithm tries to get the high frequency details and resulting image with the high resolution details looked like in Fig 4.4b

We built the training sets of images for the super resolution algorithm from band pass and high pass pairs taken from a collection of training images. Typically corresponding MxM for low frequency and NxN for high frequency patches are taken from image pairs. For more materialistic estimate of the high resolution detail, we apply the algo-

rithm four times at distributed offset relative to the patch sampling grid giving us four independent high frequency details which later we can average out.

(a) Input Image



(b) HF details predicted by algorithm

The parameter alpha regulates the matching between the low resolution patch and finding a high resolution patch that will be matches with the neighbouring patches. A rough value to start with is $0.1\,\frac{M^2}{(2N-1)}$. The affect of α on the output of the image can be seen in the next chapter.

CHAPTER 5

Experimental Results

We will see some of the results obtained by this method and compare them with the standard interpolation and independent patches algorithm. We will also see how different parameters are affecting the output of the algorithm

Figure 5.1: Input Low Resolution Image

(a) Interpolated Output

(b) Original High Resolution Image

(a) ID Algorithm Output

(b) Our Algorithm Output

Let us see their corresponding high frequency details that we tried to predict using the algorithm

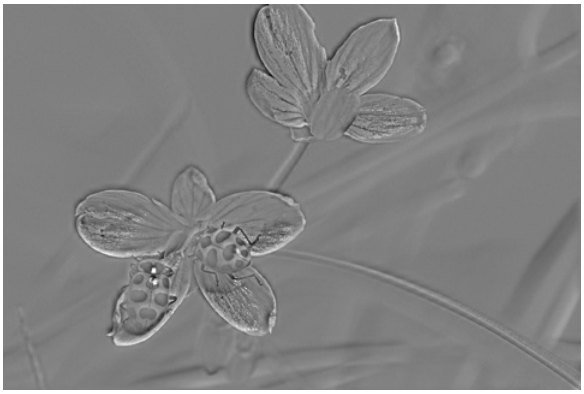
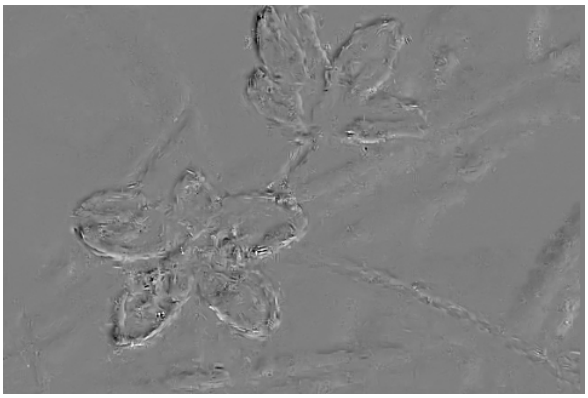


Figure 5.4: Original High Frequency Details



(a) HF predicted by ID Algorithm

(b) HF predicted by our algorithm

5.1 Bad and Good Examples

In some of the examples, the algorithm performs badly, and in some cases results are good. Let us now compare some results where the algorithm performed badly and compare their convergence plots with an image where the output is much better

(a) High Resolution Image

(b) Our Output

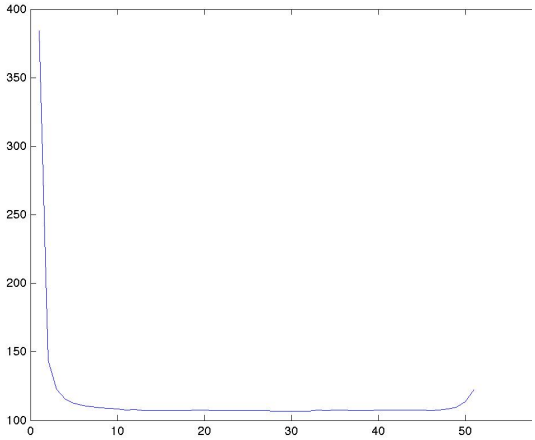
Figure 5.6: Example showing good algorithm performance

(a) High Resolution Image

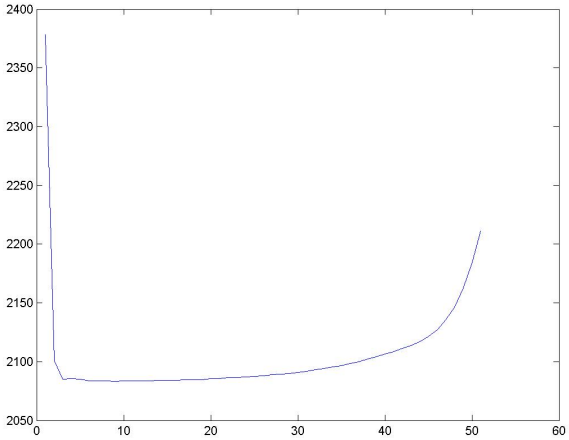
(b) Our Output

Figure 5.7: Example showing bad algorithm performance

Let us now compare their convergence plots and here we can see that for the same value of alpha, their convergence values are very different.



(a) High Resolution Image



(b) Our Output

Figure 5.8: Comparing their Convergence Plots

It should be noted that the starting values of their function itself are high, so the

value of alpha as discussed above will purely depend on patch sizes, so the problem here is due to the training data set. So training data is very important and let us see some more results where we see the influence of training data base.

5.2 Effect of Training Data Set

The approach used in the algorithm operates with patches rather than pixel values providing performance benefits. It also normalizes the training data set according to the contrast and assumes that the high frequency details of the given input image can only be predicted using only the next lower octave.

(a) Bad Training Set

(b) Good Training Set

With these two generalizations make us use this algorithm even if the training is not very similar to that of the test data set. But in that case one pass algorithm might not be sufficient. This algorithm works well if the training data resources match to the test data images we apply. Let us see some examples and inspect the quality of the output images.

(a) Bad Training Set

(b) Good Training Set

Let us take a step further and see what happens when an image in training data base is being super resolved.

(a) High Resolution Image

(b) Our Output

Figure 5.11: Image in Training Data base is being Super Resolved

Here is the convergence plot of this image

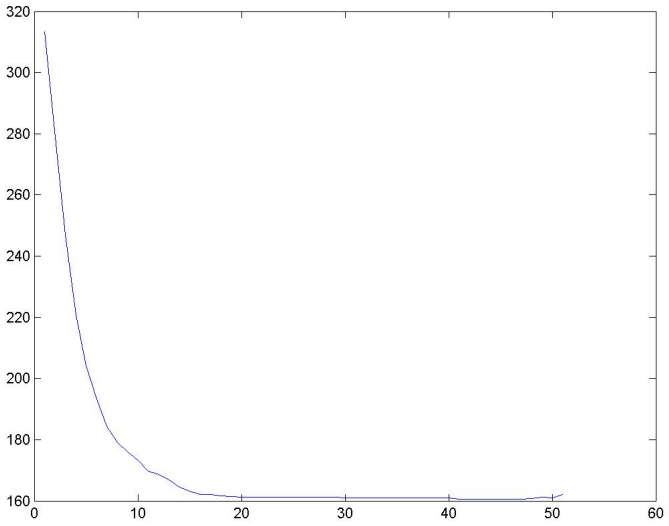


Figure 5.12: Convergence Plot

We can see that the convergence plot is not converging to zero, even though the image is in the data base, i.e the output image is not same as High Resolution Output, this is because of the merging of the patches, different methods can be used for this, two of such methods are described in the Conclusion Chapter.

5.3 Effect of Alpha parameter

The parameter α as discussed above is a trade off between smoothing and the nearest patch, now let us see some results about how this parameter is influencing the output.

(a) High Resolution Image

(b) Interpolated Output ($\alpha \approx 1$)

(a) $\alpha = .9$

(b) $\alpha = .5$

Figure 5.14: Outputs for different values of α

(a) $\alpha = .05$

(b) IID Algorithm Output ($\alpha \approx 0$)

Figure 5.15: Comparing IID Output with $\alpha = 0.05$ Output

So using this alpha as a parameter we can change the quality of the image from interpolation to IID algorithm, i.e the smooth image to non-smooth image

5.4 Effect of Patch sizes

In this section let us briefly see how patch size affect the quality of the output image and performance of the algorithm. Table 5.1 shows us the running times of the Super Resolution Algorithm for different patch sizes. It should be noted that these running times are noted on 16 GB RAM, intel i5 Machine with no application running.

Patch Sizes	**Super resolution algorithm**	**k-d tree algorithm**
Low Patchsize = 7 High Patchsize = 4	459 seconds	257 seconds
Low Patchsize = 4 High Patchsize = 3	1135 seconds	635 seconds

Table 5.1: Summary of running times with different patch sizes

So it is to be noted that patch size plays a major role in the performance of the algorithm but the quality of the output is not improved for the penalty we put on the performance. The patch sizes of four and seven for HR and LR patches respectively are working fine.

(a) Low patch = 4
High patch =3

(b) Low patch = 7
High patch =4

Figure 5.16: Comparing Outputs with different Patch sizes

5.5 Effect of Scale Factor

Till now we saw super resolving by a factor of four, so the number of pixels got increased by 16 let us apply the algorithm to increase it by two and compare the results,

(a) Factor of 2

(b) Factor of 4

Here the algorithm is able to find the missing high frequency details more clearly as much more information is available for algorithm and output image is more clear and smooth even in iid case.

(a) Factor of 2

(b) Factor of 4

(a) Factor of 2

(b) Factor of 4

It can be seen that increasing by factor of two gives much better results than increasing by four, depending on the application we can choose the factor. But in the implementation part factor value should be changed at several points both in generating training data set and super resolution algorithm

CHAPTER 6

C++ Implementation of Super-Resolution

Open CV is a cross-platform computer vision library that is developed by the open source. It can be run on Linux, Windows and Mac OS. It's light-weight but highly efficient composed of a series of C and C++ functions, it in the meantime provides the interfaces for Python, Ruby, MATLAB and other languages, thus realizing many general algorithms in the fields of image processing and computer vision

For converting the *Example based single image super-resolution* algorithm proposed by Freeman *et al.* (2002) initially MATLAB's *C coder* was used. But since the use of some MATLAB's *image processing toolbox* function, which do not have a direct conversion to C++, the output needed was not obtained. We then used *Open CV* image processing modules to implement the Algorithm.

6.1 Algorithm Parameters and Headers

headers.h

In this module all the STL and *Open CV* required header files are defined.
/*Open CV librares used*/

- **# include <opencv \ cxcore.h>**
- **# include <opencv \ cv.h>**
- **# include <opencv \ highgui.h>**
- **# include <opencv2 \ core \ core.hpp>**

/*STL librares used*/

- **# include <cmath>**
- **# include <iostream>**
- **# include <algorithm>**
- **# include <map>**
- **# include <vector>**

params.h

This header file includes all the constants and necessary parameters for the code.

- **# define patchsize_ low 7**: relates to the size of LR patch in low-res dictionary
- **# define patchsize_ high 4**: relates to the size of HR patch in high-res dictionary
- **# define intervalSize 3**: distance between two patches for the training data set
- **# define overlapSize 2** : overlap between two patches for the test data set
- **# define NN 30**: No. of nearest neighbours to be searched for k-d tree algorithm
- **# define nchannels 3**: Number of channels present for RGB images
- **# define no_ testimages 10**: Number of images for test data set
- **# define no_ trainimages 44**: Number of images for training data set
- **# define pca true**: to remove linear dependency between patches and dictionary
- **# define contrast_ norm true**

6.2 C++ Functions and Modules

imLapband.cpp

The function takes high-definition image as input. The function down-samples it, and then up-samples using `bi-cubic` interpolation and stores difference of the two images in `laplacian`. Similar operation is performed on interpolated image and result is stored in `bandpass`.

imagePatches.cpp

This function takes an image, patch and interval size as input. The function converts the image into patches of $2 * patchsize + 1$ using interval and patch size inputs and returns the output as a `vector` of images of size number of patches $\times$ size of patch.

DataGenerate.cpp

DataGenerate takes training image data set as input. This module uses the aforementioned `imLapband.cpp` and `imagePatches.cpp` functions to generate the training data set and store the output patches in low-resolution and high-resolution dictionary respectively.

ImageLumChrom.cpp

ImageLumChrom takes RGB image as input. It then converts it to YCbCr using *Open CV* function *cvtColor*. It then decomposes the image into luminance and chrominance part using in-built *Open CV* functions i.e. `split` and `merge`.

test2Patches.cpp

test2Patches takes image, size of patch of low-resolution and high-resolution, and overlap size. This function will divide the image into patches and index them using x-index and y-index. A mask matrix is generated with ones on all the indices that are converted to patches. Output is stored in a 3-d test patch matrix with size number of patches $\times$ x-index $\times$ y-index.

superResolvingAlgo.cpp

superResolvingAlgo uses the aforementioned functions i.e `ImageLumChrom`, `test2Patches` and `imLapband`. The function is not implemented completely. PCA should be implemented along with merging patches and constructing spatial compatibilities and running belief propagation algorithm.

***Open CV* Functions Used**

The reason choosing Open CV is it has many inbuilt functions in it, some of the functions which we used for implementing the code are *cvtColor*, *split*, *merge*, *add*, *subtract*, *multiply*, *GaussianBlur*, *clone*, *size*. It should be noted that even for addition and subtraction of images these function calls should be made as Open CV does not treat them as normal matrices but as images.

Implementation Problem Faced

- `cv Mat` was used as the primary image reading/writing data type. The pointer to the image matrix(or vector) is not passing between any two functions. Also the `cv Mat` is dissimilar to legacy data type `IplImage` and comparatively newer. Thus documentation for the `cv Mat` is not robust and refined as `IplImage`. This a purely implementation issue with minor hiccups.
- `meshgrid` was implemented in `C++` using nested for-loops which results in slower computational speeds. An alternative method is to use already present `cv repeat` and `cv reshape`.

CHAPTER 7

Future Work and Conclusions

In this section,we discuss about some areas that can be studied and improved which can improve the quality of the output image. As of now in the implementation standard methods are being used.

7.1 Variance Based Approach

In the two approaches discussed in the algorithm section, the first one assumed that the correlation between the patches is negligible, the other approach tried to smoothen the resultant HR image by using compatibility functions in the markov network.

Figure 7.1: IID Algorithm output

This approach is a mixture of the above two approaches. Just watch Fig 7.1, this is the output of iid algorithm, if we observe carefully, the cheek part which is very smooth is clear and the other parts are only not smooth. So the variance of the patch can be taken into account before super resolving.

Once pre-processing is done on the input image it calculates the variance of each patch, if the given patch is fairly smooth(basing on obtained value of variance), then we simply use bi-cubic interpolation to find the HR image, if the variance is high then we can try to find the patch using Belief Propagation Algorithm. Another extension of this can be using different dictionaries for different variance ranges.

7.2 Standard of Quality Evaluation

So far, all the evaluating standards applied to the evaluation of super-resolution algorithm are the statistical methods based on MSE and PSNR, but these methods are unable to reflect the quality of the image. The root mean square error from the estimated high resolution image is almost the same as bi-cubic interpolation. So such measure is not correlated with the quality of the image. Therefore, it will cause a great influence on the further study of super-resolution technology if a high-quality evaluating standard can be proposed.

7.3 Merging Patches

In the implementation of the algorithm, while we are merging the patches to from the image, different approaches can be used. We used averaging approach for merging of the patches where the patches get weights linearly from 0(edge) to 1(end of overlap). The result of that can be seen in Fig 7.2a

(a) Averaging Method

(b) Adaptive Method

Figure 7.2: Different methods for merging of Patches

We can use something called Adaptive Method, where after the high-resolution patch is partially overlapped with the left patch, the overlapping of h1 and h2 are compared by obtaining the difference value corresponding to every pixel in the overlapped part, to find the position with the smallest difference in each row. Fig 7.3 shows a clear picture of the merging of the patches. This position is the dividing position of this row, the left part left pixels belong to h1 and the right part pixels belong to h2. The result of this method can be seen in Fig 7.2b.

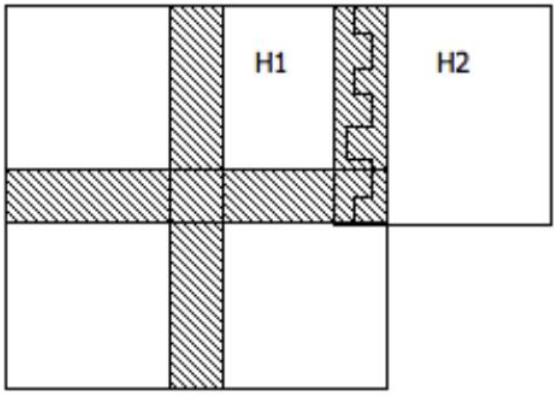


Figure 7.3: Merging of Patches

7.4 Conclusion

This approach is for enlarging single image, enlarging multiple images is different in two main aspects. There is more amount of data so multiple observations of the same pixel can be used for super resolving the images. Coherence across later frames so that the estimated high frequency details don't sparkle in the moving image. This algorithm is an instance of general training based approach that are used for super resolving. Training data sets can be used for 3-D surface shapes, removing noise etc.

REFERENCES

1. **Blake, A.**, **P. Kohli**, and **C. Rother**, *Advances in Markov Random Fields for Vision and Image Processing*. MIT Press, New York, 2011.

2. **Freeman, W. T.**, **T. R. Jones**, and **E. C. Pasztor** (2000). Markov networks for super-resolution. *34th Annual Conference on Information Sciences and Systems*.

3. **Freeman, W. T.**, **T. R. Jones**, and **E. C. Pasztor** (2002). Example-based super-resolution. *IEEE Computer Graphics and Applications*, **22**(2), 56–65.

4. **Freeman, W. T.** and **C. Liu**, *Markov Random Fields for Super-resolution and Texture Synthesis*. MIT Press, New York, .

5. **Glasner, D.**, **S. Bagon**, and **M. Irani**, Super-resolution from a single image. *In ICCV*. IEEE, 2009.

6. **Liang, L.**, **K. H. Chiu**, and **E. Y. Lam**, Fast single frame super-resolution using scale-invariant self-similarity. *In ISCAS*. IEEE, 2013. ISBN 978-1-4673-5760-9.

7. **Papoulis, A.**, *Probability Random Variables and Stochastic Processes*. McGraw-Hill, New York, 1991.

8. **Siarry, P.**, *Optimisation in Signal and Image Processing*. Wiley Press, 2009.

9. **Sudderth, E. B.** and **W. T. Freeman** (). Signal and image processing with belief propagation. *IEEE Signal Processing*.

10. **Xiong, Z.**, **D. Xu**, **X. Sun**, and **F. Wu** (2013). Example-based super-resolution with soft information and decision. *IEEE Transactions on Multimedia*, **15**(6), 1458–1465.

11. **Xu, J.**, **C. Deng**, **X. Gao**, **D. Tao**, and **X. Li**, Image super-resolution using multi-layer support vector regression. *In ICASSP*. IEEE, 2014.